Multiple Time Step Algorithms for Molecular Dynamics Simulations of Proteins: How Good Are They?

HELMUT GRUBMÜLLER,* PAUL TAVAN

Institut für Medizinische Optik, Ludwig–Maximilians-Universität München, Oettingenstrasse 67, D-80538 München, Germany

ABSTRACT: We evaluate several multiple time step (MTS) molecular dynamics (MD) methods with respect to their suitability for protein dynamics simulations. In contrast to the usual check of conservation of total energy or comparisons of trajectory details, we chose a problem-oriented approach and selected a set of *relevant* observables computed from extended test simulations. We define relevance of observables with respect to their role in the description of protein function. Accordingly, the use of quantities that exhibit chaotic behavior, like trajectory details, is shown to be inappropriate for the sake of the evaluation of methods. The accuracy of a cutoff method and of six MTS methods is evaluated, which differ in their treatment of the computationally crucial long-ranged Coulomb interactions. For each of the observables considered, the size of purely statistical fluctuations is determined to allow identification of algorithmic artifacts. The obtained ranking of the considered MD methods differs significantly from that obtained by the usual measures of algorithmic accuracy. One particular distance class method, DC-1d, is shown to be clearly superior in that no algorithmic artifacts were detected.   © 1998 John Wiley & Sons, Inc.   J Comput Chem 19: 1534–1552, 1998

Keywords: molecular dynamics simulation; protein dynamics; problem adapted accuracy; multiple time step method; relevant observables

Introduction

The function of many proteins is connected to dynamical processes within these macromolecules.[1-3] The method of molecular dynamics (MD) simulation[4-7] has recently received particular attention[8,9] due to a number of correct predictions[10-14] and is currently considered to be one of the most promising tools to describe these dynamical processes. However, the MD method exhibits two severe constraints,[15] a physical one, which excludes certain molecular properties from a proper description, and a computational one, which limits its applicability to small systems, as well as to short time scales.

The physical constraint originates from the approximate description of atomic motions in terms of *classical mechanics* by numerically solving Newton's equations for an effective interaction potential that partially models the quantum mechanics of the electronic degrees of freedom. The applicability and quality of these approximations are the subject of current discussions, but they are outside the scope of this article.

Instead, we will focus on the computational problems posed by the method of MD simulation that make up its second constraint. This constraint is set by the *enormous size* of the computational task associated with MD simulations of proteins due to the large number of particles involved (10^4–10^5), as well as the large number of integration steps that have to be carried out (e.g., 10^6 for a nanosecond simulation).

The demand to speed up simulations led to the development and application of special purpose computers[16-18] and methods to reduce the number of floating point operations required for a certain simulation time span. Examples of such methods are the conventional truncation methods[19,20] (cutoff), advanced integration schemes and multiple time stepping,[21-41] multipole methods,[42-48] as well as the grid and Ewald methods.[49-52] For reviews of these methods, see refs. 22 and 41. In this article we will ask: according to what criteria should such methods to compared, evaluated, and optimized?

At first sight it may seem that efficient MD methods should be designed to solve the equations of motion as fast *and as accurate* as possible. However, for a description of protein structure and function one generally does not have to consider *all* details of atomic motion. Due to the approximate character of molecular models, usually no physical relevance can be attributed to many of those details. Instead, one typically has to reliably compute statistical properties like mean atomic positions and fluctuations, spectra, or correlations of atomic motions. Accordingly, we will denote such quantities as *relevant*. The essence of this application-oriented approach can be summarized by the requirement that MD methods should provide accurate descriptions of relevant observables. Algorithmic accuracy in the description of irrelevant atomic details can be sacrificed.[†] Because MD algorithms generally exhibit a trade-off between efficiency and accuracy, those MD algorithms should be employed, which in that respect are as *inaccurate as possible*.

In this article we will evaluate the accuracy of six different multiple time step (MTS) methods,[25,29,33,53-55] as well as a conventional cutoff method[19] in line with the above requirements. Applying the concept of *appropriate accuracy* within the framework of MD, we evaluate the various methods by computing a set of relevant observables from extended test simulations of a simplified protein model and, subsequently, by comparing these observables with those derived from reference simulations. Another class of quite efficient methods, the fast multipole methods,[48,56,57] was studied in a previous article[57] in a similar manner.

ERROR ESTIMATION FOR MD SIMULATIONS

Evaluations of *algorithmic* accuracy generally compare selected quantities, which have been computed using the particular MD algorithm to be tested, with reference quantities that have either been computed from an MD algorithm considered to be more accurate or measured. We develop a scheme that allows the classification of procedures for the evaluation of algorithmic accuracy according to the types of quantities that they compare.

The method of MD simulation is based on *two* subsequent steps involving assumptions and approximations.

[†] Clearly, the above definition of ''relevant'' observables depends on the particular question one seeks to answer with a simulation. The crucial point here is that there are many details of the MDs that can be safely labeled irrelevant *for any physical question* one might wish to address.

The first step is the description of protein dynamics in terms of a *physical model* (see, e.g., ref. 15) defined by a large set of coupled differential equations of motion with given initial conditions. These have the detailed atomic motions, the trajectory, as their *unique and exact solution*.

A *numerical approximation*[‡] has to be computed in a second step. This task consists of the numerical integration of Newton's equations and requires a frequent determination of the forces acting on the particles. A method accomplishing the force computation and the numerical integration will be referred to as an MD algorithm.

The steps of physical modeling and numerical integration imply approximations and therefore are both subject to errors. Approximations inherent in the physical model cause discrepancies between its exact solution and the actual MD. Additional deviations of purely numerical origin are introduced at the second step. Here, the major contribution to numerical errors is typically due to an approximate treatment of long-range interactions, in particular Coulomb forces, which by far represent the computationally most extensive task of MD simulations. Accordingly, we are mainly concerned with the evaluation of those MD algorithms, which aim at an efficient computation of long-range interactions.[§]

For a further analysis we will distinguish two groups of quantities: *chaotic quantities* and *regular quantities*. This distinction is motivated by the observation that the dynamics of a protein at room temperature is chaotic: identical systems starting at almost identical positions in phase space quickly become decorrelated (i.e., well separated in phase space within a few picoseconds).[34] All quantities that share that sensitivity to slight variations of initial conditions will be termed chaotic quantities. Examples are the positions and velocities of particular atoms or the exact timings of conformational transitions. Those few quantities that do not show such chaotic behavior will be referred to as regular quantities. These include averaged quantities, such

as mean atomic positions, radii of gyration, mean temperatures, vibrational spectra, or free energy differences.

Obviously, this is not a clearcut distinction; the assignment may depend on the model employed or the duration of a simulation. However, experience shows that, at a given time scale, there *are* quantities that can be safely labeled chaotic or regular. These are the ones considered within the present context.

Applying the distinction between chaotic and regular quantities to each of the three following levels of actual protein dynamics, exact solution, and approximate solution, one obtains a total of six distinct types of quantities. These, in turn, allow six possibilities for comparison. Two of them are inaccessible, however, because they involve actual chaotic quantities, which cannot be measured. The remaining four combinations are all in common use for quality estimation.

The comparison of chaotic quantities obtained from the approximate solution with chaotic quantities obtained from the exact solution is expressed as Δ_c. Most estimates of algorithmic accuracy of type Δ_c employ comparisons of trajectories or the time development of selected atom positions or bond or dihedral angles (e.g., refs. 28, 35, 58–60). Here the deviation of the approximate solution from the exact solution is studied. An analysis of the error in the force evaluation for a hierarchical monopole approximation is given in refs. 42 and 56.

The comparison of regular quantities obtained from the approximate solution with regular quantities obtained from experiments, bridging both the physical and the numerical task is expressed as Δ_{pn}. This approach has been applied to study the effect of truncating long-range forces on the dynamical behavior of liquids[61] and proteins.[62] Here the reference quantities are observables accessible by experiment such as average atomic positions and fluctuations as well as thermodynamic properties like compressibility, specific heat, or diffusion coefficients.

The comparison of regular quantities obtained from the approximate solution with regular quantities obtained from the exact solution, bridging only the numerical task, is expressed as Δ_n. The most frequently employed test of this type is based on the total energy of the system as a regular quantity,[6, 19, 60] which is constant for conservative systems. Other test methods of the type Δ_n are based on comparisons of temperature and pressure

---

[‡] We call this task ''numerical approximation'' in order to distinguish it from the *physical* task of developing a molecular model. Here we include approximations, which rely on or are derived from specific physical properties of proteins, in our consideration.

[§] The reader might object that for MTS algorithms, for which resonance phenomena are well known,[22, 29, 41, 80] discretization errors are also of concern. But in our nomenclature these errors will be covered under a different headline: we interpret MTS methods as approximate treatments of long-range forces rather than as more coarse grained discretizations.

of a van der Waals fluid[58]; on the velocity autocorrelation function of atomic position[30]; or on radial distribution functions, compressibility, specific heat, and self-diffusion coefficients.[61]

The comparison of regular quantities obtained from the exact solution with regular quantities obtained from experiments, bridging only the physical task, is expressed as Δ_p. This type of comparison is of no direct concern within the present context, because it provides information on the quality of molecular models, not on algorithmic accuracy.

Test methods of type Δ_c are necessarily based on comparisons of irrelevant quantities. By accepting only unnecessarily accurate MD algorithms, most efficient algorithms would be rejected by such tests. Furthermore, comparisons of chaotic quantities may not even provide information on algorithmic accuracy because minute, irrelevant algorithmic deviations may be largely amplified due to the chaoticity inherent in protein dynamics. Therefore we will base our evaluations on comparisons of *regular* quantities. Recall that these arguments apply solely to the MD description of protein dynamics. For the study of other many-body systems, Δ_c results may prove quite useful, especially in cases in which the detailed trajectory is of interest (e.g., in certain astrophysical computations).

Two approaches remain to be considered: those based on Δ_n and Δ_{pn}. The latter type of comparison, however, involves quantities separated by *two* levels of approximation, the physical and the numerical ones, respectively, and therefore the opportunity to separately optimize the physical model, as well as the MD algorithm, is lost.

This disadvantage is avoided by an approach of type Δ_n: only comparisons of regular quantities obtained using the particular MD method to be evaluated with corresponding reference quantities enable a separation of artifacts caused by such an approximate method from incompatibilities of simulation results and experimental data and an optimization of MD algorithms without relying on the quality of a particular physical model. Specifically, our test simulations described below, which aim at an evaluation of approximation for the Coulomb interaction, are based on reference quantities, which were obtained by *exactly* computing that interaction, using an otherwise identical MD algorithm.

Note that regular quantities are typically averages of chaotic quantities and are therefore subject to statistical fluctuations. Accordingly, in our comparisons of type Δ_n we will have to determine the size of these fluctuations in order to separate them from the algorithmic artifacts that we wish to detect. In particular, the ensemble from which the average is taken must be sufficiently large, that, in turn, requires extended test simulations.

From the above discussion it is clear that our application-oriented approach cannot be expected to determine *the* optimal MD algorithm. Rather, problem-adapted mesures of accuracy are always defined *with respect to a particular set of quantities* and, probably, with respect to the particular physical model that has been used to carry out the test simulations. Nevertheless, test simulations on similar models should provide qualitatively similar evaluation results. Accordingly, results from test simulations on properly chosen protein model systems should actually provide information on the suitability of MD methods for the study of protein dynamics in general.

Methods

Following the above approach, we studied seven different MD algorithms, which all aim at an efficient, approximate computation of long-range forces. We proceeded in three steps.

First, a *typical* test system was selected. By typical we mean that the system should be similar enough to proteins so that it ensures relevance of our evaluation for MD studies of proteins.

Second, each algorithm was used to perform several MD simulations of the test system. Each simulation covered the time span of 1 ns. In addition, reference simulations were carried out with exact computation of long-range forces.

Finally, selected observables were computed from the obtained trajectories. As a measure of problem-adapted accuracy, these observables were compared with corresponding ones determined from the reference simulations.

TEST SYSTEM

In principle, any protein model could serve as a test system. However, to provide significant results that allow generalizations, the test system should represent a ''hard'' test case: in our example, in which we evaluate approximations for the Coulomb interaction, that interaction should sig-

nificantly contribute to the structure and dynamics of the test system.

Most quantities that are studied in typical MD studies of proteins are averages. Therefore, they are subject to statistical errors. To keep these errors small enough to enable a separation from the algorithmic deviations we intend to study, extended simulations are required. The computational effort for such extended simulation enforces the use of a *small* test system.

These criteria led us to the decision to use a simplified protein model instead of a detailed one. In a study of the low-frequency conformational dynamics of proteins, we presented that model[63] in a detailed discussion of its design and properties in that article. Here, we solely want to rehash those of its aspects that are of relevance for the present study.

The ''primary'' structure of the model consists of 100 heterogeneously charged residues. The internal structure of the residues is neglected such that the polypeptide is represented by a covalently interconnected chain of van der Waals spheres. Figure 1 shows the primary structure of the model in a stretched, unfolded conformation; the inset shows its detailed structure. The force field includes bond-stretch, bond-angle, van der Waals, and Coulomb interaction as defined in ref. 19. (Particle masses and force parameters were chosen from those of CH_2 ''extended atoms'' and of associated single bonds.) The heterogeneous charge distribution along the chain shown in Figure 1 by the bold, wavy curve was chosen to represent hydrophobic and hydrophilic interactions and, accordingly, to enable formation of a stable tertiary structure. Because, in our case, the forces that maintain the tertiary structure are exclusively of a Coulombic nature, the model should actually rep-

resent a ''tough'' test case for approximations, particularly of these forces.

Starting from the stretched configuration depicted in Figure 1, a folding process was carried out, including a subsequent relaxation and equilibration. The stability of the obtained tertiary structure on a nanosecond time scale was verified by dynamics simulations covering several nanoseconds. The resulting conformation of the protein model is depicted in Figure 2. This structure was used as the initial configuration for all test simulations.

Despite the simplicity of our model, its relevant structural and dynamical properties were shown to be quite similar to those of more realistic protein models.[63] Accordingly, the results of our test simulation should indeed provide information, whether or not the considered MD algorithms are suitable for MD studies of proteins.

TESTED MD ALGORITHMS

Test simulations on our simplified protein model were carried out using seven different MD algorithms. Each of them employs a particular approximation for the computation of the long-range

FIGURE 1. Protein model in a stretched, unfolded configuration, consisting of 100 CH_2-like ''atoms''; their partial charges are represented by the bold curve; the inset shows a zoom of the detailed structure.
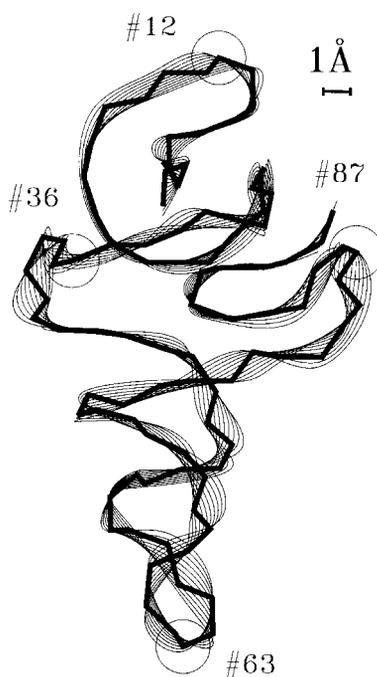
FIGURE 2. Equilibrated structure of the protein model described in the text, shown as a ''ribbon-plot''; the bold lines represent chemical bonds, four labeled circles mark atoms referred to in the text.

Coulomb interaction and aims at a reduction of the corresponding computational effort. Thus, all tested algorithms provide an approximation for the sum

$$E_{el} \propto \sum_{\langle i, j \rangle} \frac{q_i q_j}{r_{ij}} \qquad (1)$$

over all atom pairs $\langle i, j \rangle$, as well as for its spatial derivatives that are required for computation of the corresponding forces. Here E_{el} denotes the Coulomb energy of the system, q_i and q_j are the partial charges of atoms i and j, respectively, and r_{ij} denotes the distance between the two atoms. In contrast to the tested MD algorithms, the reference method, denoted by REF, performs an *exact* computation of the sum (1). Furthermore, it employs the well-known Verlet algorithm[58] for integration of the equations of motion.

Approximation of the Coulomb interaction is the key to reducing the total computational work that has to be spent in REF: the number of floating point operations necessary for evaluation of E_{el} scales with N^2, where N is the number of atoms. Because the computational effort for the short-range interaction scales linearly with N, by are most of the computational work, particularly for large systems has to spent on the determination of the Coulomb interaction.

The first approximate method we studied is the widely used "cutoff" method,[19] denoted as CUT. This method *neglects* pair interaction of atoms separated further than a certain distance, the cutoff radius r_{cut}. The cutoff is achieved by multiplying the interaction energy with a "switching function" $f(r)$, which is unity for $r = 0$ and reduces the forces to zero, usually in a continuous differentiable manner. Accordingly, the sum in eq. (1) is approximated by

$$E_{el} \approx \sum_{\langle i, j \rangle_{cut}} f(r_{ij}) \frac{q_i q_j}{r_{ij}}.$$

Due to the switching function only a small subset $\langle i, j \rangle_{cut}$ of all atom pairs $\langle i, j \rangle$ remains to be considered, namely, those for which $r_{ij} < r_{cut}$. Commonly used values for r_{cut} are in the range of 8–15 Å. In our simulation we used the switching function[19] $f(r) = (1 - r^2/r_{cut}^2)^2$ and a cutoff radius r_{cut} of 10 Å.

The other six algorithms investigated were *distance-class* (DC) methods. By approximately accounting for pair interaction beyond r_{cut}, they are designed to avoid artifacts caused by the cutoff-type neglect of these interactions. Nevertheless, they achieve a comparably large reduction of the computational effort by employing an MTS scheme. That scheme rests on the observation that the time derivative of pair interactions decreases with increasing distance r_{ij}. Accordingly, during numerical integration, the slowly fluctuating, long-range force contributions need not be explicitly computed as often as the rapidly varying short-range interactions. Instead, the long-range forces can be estimated by extrapolation procedures at many intermediate integration steps.

The frequency of explicit force computation is defined for each atom pair by its assignment to a particular DC, which is chosen according to r_{ij} as follows: let $\{R_0, \ldots, R_n\}$ be a set of radii with $R_j < R_{j+1}$ for all $j = 0, \ldots, n - 1$ and $R_0 = 0$. Then the set of atom pairs $\langle i, k \rangle$ that satisfy $R_j \leq r_{ik} < R_{j+1}$ is called DC j. Using eq. (1) and the respective spatial derivatives, the force $\mathbf{F}^{(j)}$ acting on a particular atom and originating from all other atoms within DC j is computed explicitly once every $n := 2^j$ integration steps. All other forces are estimated from previously computed forces according to the linear extrapolation formula

$$\mathbf{F}_{i+kn}^{(j)} \approx a_i^{(j)} \mathbf{F}_{kn}^{(j)} + b_i^{(j)} \mathbf{F}_{(k-1)n}^{(j)}, \qquad (2)$$

where $i = 0, \ldots, n - 1$ and $i + kn$ is the number of the current integration step. The principles of this method and their justification are outlined in ref. 29. For our simulations, we used four DCs covering the intervals (in Å): $[0 \cdots 4]$, $[4 \cdots 7]$, $[7 \cdots 11]$, and $[11 \cdots \infty]$.

The six DC algorithms considered here, DC-0, DC-i, DC-1a, DC-1b, DC-1c, and DC-1d, respectively, differ in the choice of the "extrapolation coefficients" $a_i^{(j)}$ and $b_i^{(j)}$. Table I lists the extrapolation coefficients that we used. Here, i is the number of integration steps carried out since the most recent exact force computation of atom pairs within class j. As an illustration, Figure 3 shows for all six DC algorithms under consideration the time development of extrapolated forces (vertical lines) that enter into the integration of the Newtonian equations of motion. The corresponding exact forces are drawn as bold lines.

Method DC-i (impulse), which is closely related to the RESPA algorithm proposed in ref. 30, was

TABLE I.
Extrapolation Coefficients $a_i^{(j)}$ and $b_i^{(j)}$ Used for Six Distance-Class Algorithms.

Algorithm	$a_i^{(j)}$	$b_i^{(j)}$
DC-i	n for $i = 0$, 0 otherwise	0
DC-0	1	0
DC-1a	$6n \dfrac{(n^2 - n + 1) - i(n - 2)}{(n + 1)(2n^2 + 1)}$	$\dfrac{2(-2n^2 + 3n - 1) + 6i(n - 1)}{2n^2 + 1}$
DC-1b	$1 + i/n$	$-i/n$
DC-1c	$(n + 1)/2$ for $i = 0$, 1 otherwise	$(1 - n)/2$ for $i = 0$, 0 otherwise
DC-1d	$\dfrac{3n^2 - 2n + 1}{n(n + 1)} - 3i\dfrac{n - 1}{n(n + 1)}$	$1 - \dfrac{3n^2 - 2n + 1}{n(n + 1)} + 3i\dfrac{n - 1}{n(n + 1)}$

We used the abbreviation $n := 2^j$; i is the number of integration steps carried out since the most recent exact force computation for class j.

derived in ref. 29 (called VERLET-I therein) using the so-called "Verlet criterion" that ensures equivalence to exact forced computation in those cases in which only one DC (not necessarily the innermost) is populated by atom pairs. In fact, no force extrapolation occurs here. Rather, all forces from n subsequent integration steps are combined and applied instantaneously (cf. the large spikes depicted in Fig. 3, DC-i). Because the magnitude of these forces increases exponentially with the number of DCs, DC-i tends to cause severe artifacts if more than three DCs are employed.[29]

Method DC-0, first applied to MD simulations in ref. 27 (see also ref. 64), avoids this disadvantage. Here, exact forces computed every nth integration step are constantly applied within all subsequent $n - 1$ integration steps. This can be considered a 0th order extrapolation, which appears as stair steps in Figure 3, DC-0.

The above two methods estimate forces on the basis of the most recently computed force; all $b_i^{(j)}$ vanish in these cases. Better approximations may be obtained if, in addition to the most recently computed exact forces, "older" forces are considered, which have been computed n steps before.[29] In this case, many possible choices for the extrapolation coefficients exist. The most obvious choice, which to our knowledge has not yet been used for MD simulations, is a linear extrapolation (DC-1b). The corresponding picture in Figure 3 shows the resulting force development in time.

However, method DC-1b does not obey the Verlet criterion.[29] Among all sets of extrapolation coefficients, which *do* fulfill the Verlet criterion, those that deviate as little as possible from a linear extrapolation constitute method DC-1a, as derived in ref. 29. Note that, despite this minimal principle, the coefficients differ considerably from the ones in DC-1b: in contrast to the linear extrapolation, here the a coefficients decrease whereas the b coefficients rise. This fact becomes apparent in Figure 3, DC-1a, where increased discontinuities in the force development can be observed.

A third set of coefficients (DC-1c) was proposed by Skeel and Biesiadecki.[65,66] This method also obeys the Verlet criterion. In addition, one can show that DC-1c minimizes the orce discontinuities (not shown in Fig. 3) that arise if atom pairs move from one DC to another in the course of a simulation.[34] However, this method may suffer
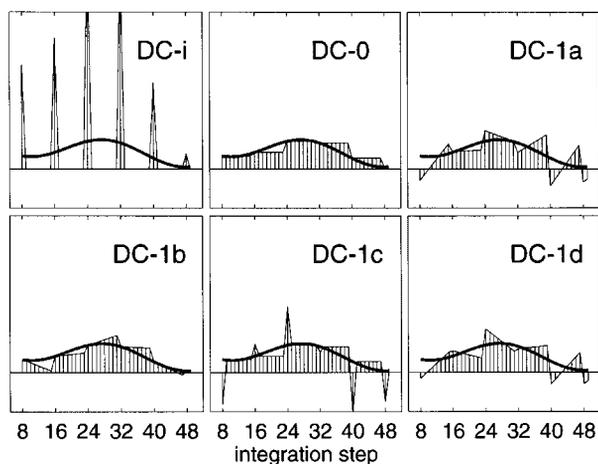
FIGURE 3. Extrapolated forces (vertical lines) for distance class $j = 3$, as defined for the six distance-class algorithms referred to in the text [cf. eq. (2), Table I]. For the numerical integration of the Newtonian equations of motion, these extrapolated forces are used as an approximation for the exact forces (bold lines).

from artifacts, which are smaller than, but of similar origin as those apparent in DC-i (cf. the corresponding pictures in Fig. 3, where sharp peaks are visible).

To combine the advantages of the above algorithms while avoiding possible resonant artifacts, we propose a new method DC-1d, which is the last method considered here. Although it deviates only slightly from method DC-1a, we expected it to be superior, because it additionally avoids, like DC-1c, force discontinuities caused by atom pairs crossing DC boundaries. This algorithm is derived and discussed in detail in ref. 34.

TEST SIMULATIONS

A total of 232 test simulations, each of one nanosecond duration, were carried out using the eight MD algorithms described above. Integration step sizes of 0.5, 1.0, and 2.0 fs were employed.[¶] All 232 simulation started with almost identical initial conditions, which were derived from the model protein structure described above by applying minute random modifications of the order of 10^{-6} Å to atomic positions. The chaoticity inherent in the dynamics leads to a rapid decorrelation of the trajectories after a few picoseconds and therefore guarantees that the 232 simulation are essentially independent of each other.

A described elsewhere,[63] our model system was found to exhibit several conformational states, between which rare transitions occur on a time scale of several hundred picoseconds. That conformational dynamics, although initially unexpected by us, resembles a ubiquitous property of low-frequency protein dynamics[67, 68]: in many cases macroscopic observables have been found to be strongly influenced by the underlying conformational state (see, e.g., refs. 69, 70). It was not our intention to demonstrate that well-known influence of molecular conformation on a variety of observables. Instead, we wanted to concentrate on the influence of different approximation schemes on the *dynamics* of proteins in those cases in which the molecular structure is *unaffected* by these approximations. Therefore, those trajectories were selected from our test simulations for further analysis, that did *not* exhibit conformational transitions.

[¶] For the MTS algorithms this choice implies force update periods of 1, 2, and 4 fs for the second distance class ($i = 1$); of 2, 4, and 8 fs for the third distance class ($i = 2$); and of 4, 8, and 16 fs for the fourth distance class ($i = 3$).

The system was considered to stay within the initial conformation state if the average structure in the test simulation did not considerably deviate from the reference structure and if no jumps of selected atomic distances occurred.

The first criterion was assessed by computing the *maximum* deviation of mean atomic positions from those of the reference simulation. This deviation had to be smaller than 2.5 Å. For the second criterion we selected four atoms, #12, #36, #63, and #87, the location of which within the molecular structure is indicated by four labeled circles in Figure 2. Because conformational changes during the simulation could be monitored by fast, significant changes of the distances among these four atoms, we selected only those simulations that did not exhibit such distance jumps.

The above selection excludes those trajectories from our analysis in which deviations from the reference simulation are simply caused by occasional conformational transitions or by *structural* changes induced by algorithmic artifacts. Therefore, any observed significant deviation actually reflects an artifact concerning the *dynamics* of our system. [A similar approach was taken in ref. 57, where a more detailed protein model was used.]

RELEVANT QUANTITIES

For our application-oriented evaluation of the seven MD algorithms sketched above, we chose a set of regular quantities that enable comparisons of the type Δ_n. Our choice was guided by the requirement that the respective regular quantities should be useful for explanatory purposes in MD studies of protein function: these quantities should be *relevant*. As explained in detail below, these quantities include atomic fluctuations, vibrational spectra, autocorrelation functions, and cross-correlations of atomic motions. We also considered configuration space density distributions due to their close relation to free energy computation and reaction rates. For comparison, we included in our analysis the drift of total energy as a frequently employed measure of algorithmic accuracy, although in our view that quantity is of no particular functional relevance.

Despite the considerable length of the test simulations, certain observables exhibited statistical errors. To separate deviation caused by the particular approximation scheme (''algorithmic deviations'') from statistical errors, the latter were

estimated by comparing independent *reference* simulations with each other.

Note the following abbreviations: $\mathbf{r}_s(t_i)$ is the position of atom s at time t_i at the ith integration step in the simulation[‖]; $\langle \mathbf{r}_s \rangle := \sum_{t_i} \mathbf{r}_s(t_i)/N_t$ is the average position of atom s, determined from a trajectory of N_t coordinate sets (i.e., the number of integration steps); and length $T = 1{,}048{,}576$ fs; the integration step size is Δ_t; the superscript ref denotes an observable computed from a reference simulation.

Drift of Total Energy

Due to algorithmic noise forces, we expected some of the algorithms to cause considerable energy drifts, amounting to several hundred kilocalories/mole during a 1-ns simulation. Such drift would imply a large change of temperature and prevent further comparisons. Therefore, we decided to compensate energy drifts by a weak coupling to a heat bath. Accordingly, the energy flow to or from the system was used as a measure for the algorithmic heat production and, hence, total energy drift.

Average Spectrum of Atomic Vibrations

The spectrum $\hat{\mathbf{r}}_s(\omega)$ of atomic vibrations, derived by a Fourier transform of the trajectory of atom s, determines the contribution of the motion of atom s to the infrared spectrum of a protein. For our evaluation of algorithm is accuracy, we used the average power spectrum $S(\omega)$ defined by

$$S(\omega) := \sum_s |\hat{\mathbf{r}}_s(\omega)|^2. \tag{3}$$

Spectra were computed for two different frequency ranges, a high-frequency range (0.5–250.0 ps^{-1}) and a low-frequency range (0.001–4.0 ps^{-1}).

High-Frequency Spectra of Atomic Vibrations. For the computation of the high-frequency spectra, each 1-ns trajectory was partitioned into $M = 511$ consecutively overlapping segments $\{\mathbf{r}(m\Delta_M), \ldots, \mathbf{r}((m+2)\Delta_M)\}$, $m = 0 \cdots M - 1$, of length $2\Delta_M = 4096$ fs each. According to the procedure described in ref. 71, all segments were multiplied with a Hanning window function and then subject to a Fourier transform. From the resulting M spectra an average was taken.

[‖] For technical reasons, all sums over s take only every fifth atom along the chain into account: $s = 1, 6, 11, \ldots, 91, 96$.

Low-Frequency Spectra of Atomic Vibrations. The low-frequency Fourier transforms were computed according to

$$\hat{\mathbf{r}}_s(\omega) = \sum_{k=0}^{N_{\text{sm}}-1} \bar{\mathbf{r}}_s(t_k)\exp(i\omega t_k), \quad t_k = k\sigma,$$

with smoothened trajectories $\bar{\mathbf{r}}_s(t)$,

$$\bar{\mathbf{r}}_s(t) = \frac{1}{Z} \sum_{k=-10\sigma/\Delta_t}^{10\sigma/\Delta_t} \mathbf{r}_s(t)\exp\left[-\frac{(t-k\Delta_t)^2}{2\sigma^2}\right],$$

$$Z = \sum_{k=-10\sigma/\Delta_t}^{10\sigma/\Delta_t} \exp\left[-\frac{(t-k\Delta_t)^2}{2\sigma^2}\right].$$

The sampling rate of the smoothened trajectory was chosen in correspondence to the width of the smoothing window, $\sigma = 128$ fs. $N_{\text{sm}} := T/N_t$ denotes the number of smoothened coordinate sets.

Atomic RMS Fluctuations

The average displacement σ_s of an atom s from its average position $\langle \mathbf{r}_s \rangle$ was computed by

$$\sigma_s = \left[\sum_{t_i} \left((\mathbf{r}_s(t_i) - \langle \mathbf{r}_s \rangle)^2 \right)/N_t \right]^{1/2}. \tag{4}$$

Here σ_s is a measure for atomic mobility and usually depends on the molecular environment of the particular atom under consideration: atoms, for example, that belong to solvent-exposed side groups of proteins, typically show a higher mobility than backbone atoms.[72] Because mobilities can be determined by X-ray or neutron scattering experiments, that observable is frequently used as a check for the quality of molecular models.

Correlations of Atomic Motions

Correlation function serve to characterize atomic motions. Autocorrelation functions of atomic positions or velocities, on the one hand, provide information on the diffusive character of atomic motion and enable the determination of friction coefficients.[72] Cross-correlations of atomic motions, on the other hand, serve to detect collective motion and thus reveal functional, probably causal, interrelations between distant parts of a protein. Due to the widespread use of these quantities, we studied displacement autocorrelation function $C_s^{\text{dis}}(\tau)$ and velocity autocorrelation function $C_s^{\text{vel}}(\tau)$, as well

as cross correlation functions $K_{s_1 s_2}$, as relevant quantities. They were computed according to

$$C_s^{\text{dis}}(\tau_j) = \frac{\sum_{i=0}^{N_j-1}\left(\mathbf{r}_s(t_i) - \langle\mathbf{r}_s\rangle\right)\left(\mathbf{r}_s(t_i + \tau_j) - \langle\mathbf{r}_s\rangle\right)}{\sum_{i=0}^{N_j-1}(\mathbf{r}_s(t_i) - \langle\mathbf{r}_s\rangle)^2},$$

(5)

$$C_s^{\text{vel}}(\tau_j) = \frac{\sum_{i=0}^{N_j-1}\dot{\mathbf{r}}_s(t_i)\dot{\mathbf{r}}_s(T_i + \tau_j)}{\sum_{i=0}^{N_j-1}\dot{\mathbf{r}}_s(t_i)^2},$$

(6)

$$K_{s_1 s_2} = \frac{k_{s_1 s_2}}{(k_{s_2 s_2} k_{s_2 s_2})^{1/2}},$$

(7)

with the covariance matrix

$$k_{s_1 s_2} = \sum_{t_i}\left(\mathbf{r}_{s_1}(t_i) - \langle\mathbf{r}_{s_1}\rangle\right)\left(\mathbf{r}_{s_2}(t_i) - \langle\mathbf{r}_{s_2}\rangle\right).$$

In the above equations, $N_j = (T - \tau_j)/\Delta_t$ and $\tau_j = j\Delta_t$; the overdot denotes the time derivative.

Projected Configuration Space Density

A fundamental thermodynamic quantity of many-body systems is the phase space density $\rho(\mathbf{r}^N, \mathbf{p}^N)$, which is generated by their dynamics. Here, $\mathbf{r}^N$ and $\mathbf{p}^N$ denote the $3N$-dimensional vectors of all N atomic position and momenta, respectively; ρ is rarely used and is determined on its own, but it is closely related to statistical observables like entropy, free energy, or reaction rates. An approximation for ρ can be computed using an ensemble, which is generated by means of an MD simulation, by dividing phase space into volume elements V^k: for long simulation times the fraction of trajectory points within V^k approaches phase space density at the location of V^k.

However, due to the limited length of MD simulations, as well as due to the high dimensionality of phase space, such a density estimate generally does not represent a regular quantity. Even for our small test system the dimension of phase space is $6N = 600$; therefore, the number K of volume elements V^k is extremely large even for coarse resolutions, thus, the average number of points per volume element is small. Correspondingly large statistical errors inhibit a straightforward use of phase space density as a measure for algorithmic accuracy.

However, if instead of taking *all* $6N$ degrees of freedom into account, only a *few* degrees of freedom are considered, regular quantities can be derived. For this purpose one projects ρ onto a low-dimensional subspace spanned by M degrees of freedom, $\mathbf{c} := \{c_i(\mathbf{r}^N, \mathbf{p}^N)\}$, $i = 1, \ldots, M$, and obtains a *projected phase space density* $\rho_c(\mathbf{c})$:

$$\rho_c(\mathbf{c}) = \int d\mathbf{r}'^N d\mathbf{p}'^N \rho(\mathbf{r}'^N, \mathbf{p}'^N)\delta(\mathbf{c} - \mathbf{c}').$$

The density $\rho_c(\mathbf{c})$ can be considered a *regular* quantity if the number of trajectory points per volume element *within the subspace* is large enough to allow a statistical analysis. We restrict our discussion to projections onto *conformational* degrees of freedom, $\mathbf{c} = \{c_i(\mathbf{r}^N)\}$, usually referred to as "conformational coordinates."[73] For a discussion of their central role in the statistical mechanics of conformational transitions, see ref. 74. In particular, the strict definition of conformational substates in terms of minima of free energy as a function of suitably projected configuration space densities led to an MD based method, "conformational flooding," to predict microsecond conformational transitions.[71]

For our test simulations, we selected projections onto a number of $P = 50$ 2-, 3-, and 4-dimensional conformational subspaces, defined by pairs, triples, and quadruples of interatomic distances, respectively. For the 2-dimensional case, we used distance pairs $\mathbf{c} = (d_{n_1 n_2}, d_{n_3 n_4})$ among four selected atoms, n_1, n_2, n_3, and n_4, respectively. Figure 2 shows the locations of the four selected atoms (#12, #36, #63, and #87) within the protein model. Projections onto all 15 2-dimensional subspaces, which can be defined using the six distances $d_1 = d(\#12; \#36)$, $d_2 = d(\#12; \#63)$, $d_3 = d(\#12; \#87)$, $d_4 = d(\#36; \#63)$, $d_5 = d(\#36; \#87)$, and $d_6 = d(\#63; \#87)$, were computed, yielding a total of 15 different density distributions. Twenty 3- and 15 4-dimensional projections were similarly defined.

The decision to use projections defined by interatomic distances was inspired by the possibility of computing conformation-controlled reaction rates from such projected densities and, corresponding, of explaining the kinetics of protein function on the basis of MD simulations. As an example, consider the hypothetical docking process of two enzymes A and B depicted in Figure 4. That process exhibits similarities to the docking reactions of G proteins (e.g., transducin)[75] to photoreceptors. The two enzymes are supposed to have three docking sites. Their mutual distances are d_{12}, d_{23}, D_{12}, and D_{23}, as shown in Figure 4. For simplicity, we assume enzyme A to be flexible, such that d_{12} and d_{23} show considerable fluctuations, and enzyme B to be rigid, such that D_{12} and D_{23} are constant.

These properties are indicated by the shapes of the two enzymes. As the system moves, the probability of the occurrence of a docking reaction will be large if all three active sites simultaneously fit together, that is, if $d_{12} \approx D_{12}$ and $d_{23} \approx D_{23}$; otherwise the docking probability will be small. Hence, the corresponding reaction rates depend on the frequency of distance matchings and is thus determined by the projected configuration space density $p(d_{12}, d_{23})$ in the vicinity of location (D_{12}, D_{23}).

A similar approach was employed in ref. 76, where electron transfer rates within a protein were computed from MD simulations. In that work, following Marcus theory,[77] the probability of an electron transfer at each instance of time was assumed to be large whenever the corresponding change of electrostatic energy was small. Because that energy difference is a function of atomic positions, the obtained transfer rate is determined by the projected phase space density onto a conformational subspace that in this case is a 1-dimensional one.

The above examples illustrate that projected configuration space densities can be considered to be a relevant quantity for a wide range of MD applications.

For each algorithm a and each test simulation, we determined ρ_c by extracting ensembles S_a consisting of a total of $N_c = 131{,}072$ configurations. For this purpose, each of the $P = 50$ subspaces was subdivided into K rectangular volume elements $V^{k,p}$, $k = 1 \cdots K$, $p = 1 \cdots P$. The subdivisions were set up in such a way that in case of the ensemble R_1 obtained from the reference simulation, all rectangles $V^{k,p}$ contained an identical

number $N_{R_1} := N_c / K$ of configurations. As an illustration, Figure 5 shows such a subdivision into $K = 16$ rectangles $V^{16,p}$ that are separated by bold lines. The ensemble R_1 is depicted as a cloud of dots, each dot representing one particular configuration.**

Using this subdivision, deviations $\Delta^p_{S_a - R_1}$ of the projected phase space density from the reference density were computed by

$$\Delta^p_{S_a - R_1} = \left[\frac{1}{K-1} \sum_{k=1}^{K} \left(\frac{N^{k,p}_{S_a} - N_{R_1}}{N_{R_1}} \right)^2 \right]^{1/2}. \quad (8)$$

where $N^{k,p}_{S_a}$ is the number of configurations out of ensemble S_a that fall into rectangle $V^{k,p}$ for projection p.

Nonvanishing deviations $\Delta^p_{S_a - R_1}$ may result from statistical fluctuations of the finite configuration counts $N^{k,p}_{S_a}$ or be caused by the particular MD algorithm to be tested. To decide whether observed deviations significantly point to an algorithmic artifact, the purely statistical contribution of $\Delta^p_{S_a - R_1}$ has to be estimated. [Note that these statistical errors of the $N^{k,p}_{S_a}$ must be expected to be considerably larger than $(N^{k,p}_{S_a})^{1/2}$, because subsequent configurations, temporally separated by $T/N_c = 8$ fs, are strongly correlated and therefore do not represent *independent* random events.]

We estimated the fluctuations by comparing ensemble R_1 with a second reference ensemble,

** For technical reasons, only a fraction of all N_c configurations is shown.

FIGURE 4. Schematic enzymatic docking process to illustrate the functional relevance of configuration space density projections; an explanation is given in the text.
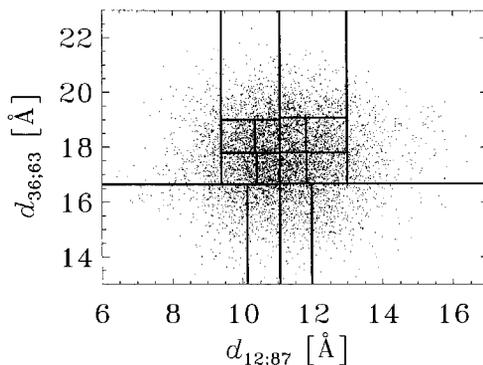
FIGURE 5. Ensemble R_1 of configurations generated by a reference simulation of the protein model. Points in configuration space, projected onto the 2-dimensional subspace defined by two interatomic distances $d_{12;87}$ and $d_{36;63}$, are represented by dots; a subdivision $V^{16,p}$ of this subspace (see text) is depicted using bold lines.

R_2. For the second reference simulation the initial atomic velocities were slightly varied to generate an independent trajectory. Thus, deviation $\Delta_{R_2-R_1}^p$ should solely result from statistical fluctuations.

Based on that assumption, the average deviation of the two reference ensembles

$$\langle \Delta_{R_2-R_1} \rangle := \left(\sum_{p=1}^{P} \Delta_{R_2-R_1}^p \right) \Big/ P \qquad (9)$$

was used to estimate the statistical contribution to the observed deviations. As a measure for the error of that estimate we used the variance $\sigma_{R_2-R_1}$ of the $\Delta_{R_2-R_1}^p$,

$$\sigma_{R_2-R_1} := \left[\frac{1}{P-1} \sum_{p=1}^{P} \left(\Delta_{R_2-R_1}^p - \langle \Delta_{R_2-R_1} \rangle \right)^2 \right]^{1/2}. \qquad (10)$$

Similarly, for each algorithm to be tested, a separate $\sigma_{S_a-R_1}$ was computed on the basis of deviations $\Delta_{S_a-R_1}^p$. The significant contribution to the average deviation $\langle \Delta_{S_a-R_1}^p \rangle$, the algorithmic accuracy Δ_a, can now be readily obtained by

$$\Delta_a = \langle \Delta_{S_a-R_1} \rangle - \langle \Delta_{R_2-R_1} \rangle \pm \sigma_a \qquad (11)$$

with a certain error range σ_a.

If the $P = 50$ phase space projections were independent of each other, the error range σ_a could be estimated by

$$\sigma_a \approx \sigma_a^{\min} := \left[\left(\sigma_{R_2-R_1}^2 + \sigma_{S_a-R_1}^2 \right) / P \right]^{1/2}. \qquad (12)$$

However, because statistical independence cannot be assumed, one has to consider the upper limit

$$\sigma_a^{\max} := \left[\left(\sigma_{R_2-R_1}^2 + \sigma_{S_a-R_1}^2 \right) \right]^{1/2}. \qquad (13)$$

Then one may safely assume σ_a to be in the range between $\sigma_a^{\min}$ and $\sigma_a^{\max}$.

It is not easy to determine *a priori* which level of coarse graining should be used for our comparisons of ρ_c: few large volume elements, on the one hand, minimize the statistical fluctuations $\langle \Delta_{R_2-R_1} \rangle$ but provide no detailed picture of the density distribution. Many small volume elements, on the other hand, provide a high resolution picture of the density distribution but suffer from large statistical fluctuations. To provide an unbiased analysis, we compared projected phase space densities using various resolutions (i.e., various numbers K of volume elements $V^{k,p}$).

Results

The 232 test simulations of 1-ns duration each were analyzed. According to the criteria outlined further above, we selected for each of the eight MD algorithms (REF, CUT, DC-i, DC-0, DC-1a, DC-1b, DC-1c, and DC-1d) and for each of the three integration step sizes of 0.5, 1.0, and 2.0 fs, respectively, a suitable-1−ns trajectory for further analysis. Wherever appropriate, results of all 24 trajectories are shown, but in most cases only the eight simulations obtained with a 1.0-fs step size are considered.

DRIFT OF TOTAL ENERGY

Figure 6 shows the drift of total energy per picosecond for all 24 1-ns simulations. One can observe that, for each of the algorithms, the energy drift increases with the integration step size. CUT does not significantly affect conservation of total energy that quite accurately results in the case of the reference simulation, REF. In contrast, all DC algorithms are associated with considerable heat production. Without coupling to a heat bath, that heat production would cause a denaturation of the system with a 1-ns simulation.

The large energy drift observed for the DC algorithms is due to errors associated with the force extrapolations, as illustrated in Figure 3. These deviations from the exact forces represent "noise" forces, which transfer kinetic energy into the molecular system. Note that the magnitude of noise, which may be estimated by inspecting the force deviations apparent in Figure 3, does not
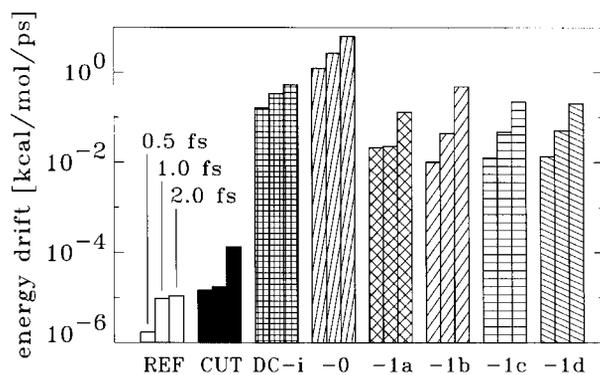
FIGURE 6. Drift of total energy per picosecond for the reference simulation REF, as well as for the seven investigated algorithms. All eight algorithms were applied with integration step sizes of 0.5, 1.0, and 2.0 fs.

completely correspond to the observed energy drift in Figure 6: large noise may be expected for DC-i, DC-1c, DC-1a, or DC-1d; however, the latter three algorithms that all obey the Verlet criterion exhibit a moderate energy drift only. In contrast, the largest energy drift is caused by the relatively small noise forces of DC-0. One may conclude that the Verlet criterion serves to reduce the total energy drift caused by algorithmic noise forces in DC methods, in line with the objectives that led to the formulation of that criterion.

AVERAGE HIGH-FREQUENCY SPECTRUM OF ATOMIC VIBRATIONS

As is apparent in Figure 3, the force discontinuities caused by DC algorithms occur at every $n = 2^j$th integration step in our simulations. Accordingly, their influence on the atomic motion should become apparent in averaged Fourier spectra of atomic vibrations at *high* frequencies (Fig. 7). We computed these spectra according to eq. (3) for the integration step sizes of 0.5 and 1.0 fs, respectively. The two spectra obtained from REF are plotted as dashed lines in the upper left picture in Figure 7. Due to the absence of algorithmic noise, they vanish at frequencies above 100 ps^{-1}. In contrast, the corresponding two spectra obtained from DC-i (solid lines) exhibit sharp artificial resonance peaks at 250 ps^{-1} and, in the case of 1.0-fs step size, additionally at 125 ps^{-1}. These are the frequencies expected for the algorithmic noise forces, because they represent the durations of four and eight integration steps, respectively.

The upper right plot in Figure 7 shows the same data, this time plotted as the relative error of DC-i with respect to REF, again using 0.5-fs, as well as 1.0-fs, integration step sizes. In a corresponding fashion, the bottom four plots of Figure 7 show reltaive errors associated with DC-0, DC-1a, DC-1b, and DC-1c, respectively. Note, however, the diminished error scales caused by the reduced force discontinuities in these algorithms. Not shown are high-frequency spectra of CUT, as well as DC-1d: the former is identical to the reference spectrum, whereas the latter spectrum is similar to DC-1c. Due to the extended length of the simulations, no statistical fluctuation could be observed in the high-frequency spectra. For all methods, the magnitudes of the algorithmic noise peaks match the sizes of the force discontinuities illustrated in Figure 3 quite well.

Are these partially pronounced artifacts relevant? For an answer, first note the peak at
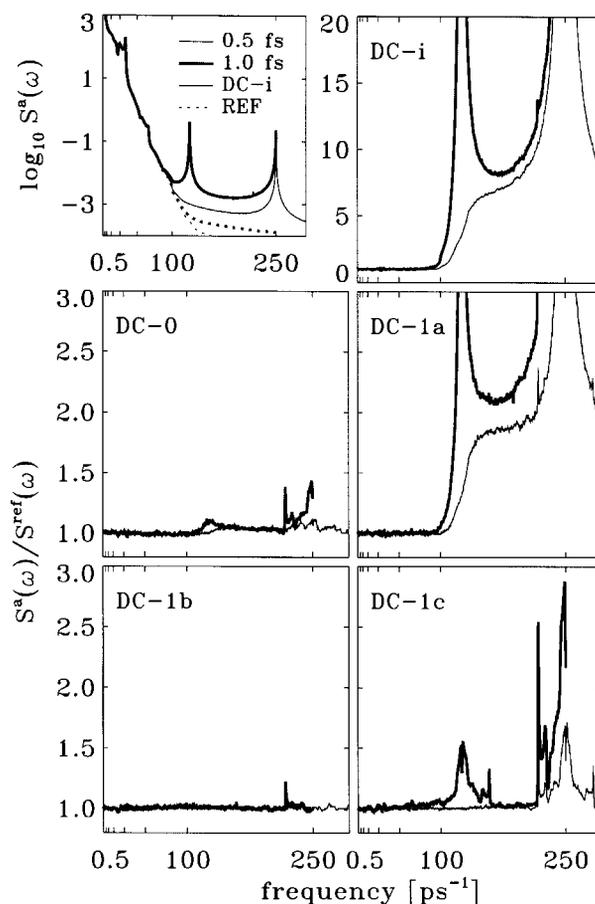
FIGURE 7. High-frequency spectra of atomic mobilities; the upper left shows the reference spectrum (dashed lines) and DC-i spectrum (solid lines) that were both obtained with integration step sizes of 0.5 fs (thin lines) and 1.0 fs (bold lines); the remaining graphs show the relative errors S^a / S^{ref} of the spectra S^a obtained using the algorithms $a \in \{$DC-i, DC-0, DC-1a, DC-1b, DC-1c$\}$ with integration step sizes of 0.5 fs (thin lines) and 1.0 fs (bold lines); note the different scale for graph DC-k in the upper right; no significant relative error was found for CUT (results not shown); the relative error of DC-1d (not shown) is similar to that of DC-1c.

$\approx 30 \text{ ps}^{-1}$ in the upper left graph of Figure 7, which originates from bond stretch vibrations. These represent the fastest *physical* degrees of freedom in the system. Therefore, at higher frequencies, the spectrum does not represent physical properties. Because no deviations of the spectra are observed in the frequency range below 30 ps^{-1} in which physical properties are monitored, the apparent algorithmic artifacts are of no *direct* concern. It remains to be clarified, however, whether the high-frequency noise and the necessary compensation of the energy drift cause errors in the

computation of relevant quantities. All further comparisons serve to answer that question.

AVERAGE LOW-FREQUENCY SPECTRUM OF ATOMIC VIBRATIONS

We first focus on low-frequency atomic motions: our extended simulations permitted the computation of low-frequency spectra, including frequencies as low as 0.05 ps^{-1} with small statistical error. The upper part of Figure 8 shows three spectra, each of which was obtained by averaging over the three simulations with integration step sizes 0.5, 1.0, and 2.0 fs. The dashed line represents the reference spectrum.

None of the DC algorithms shows significant deviations from REF; the spectrum obtained from algorithm DC-1d is shown as an example (thin, solid line). ON the contrary, CUT causes a pronounced suppression of atomic mobilities at frequencies below 0.2 ps^{-1}. Inspection of the relative deviations, S^a/S^{ref} (bottom of Fig. 8) reveals a cutoff-induced suppression of up to 40% (bold line) with respect to REF. A noteworthy consequence of this finding is that apparently long-range interaction above 10 Å, which are absent in CUT but present in REF, have a significant impact on the low-frequency dynamics of our protein model. In that respect our model reproduces well-known properties of more realistic protein models.

ATOMIC RMS FLUCTUATIONS

To reduce statistical errors, the above spectra were averaged over all atoms [cf. eq. (3)], thereby probably hiding algorithmic artifacts related to motion of individual atoms. For a more detailed measure of accuracy, we computed the atomic mobilities σ_s as defined in eq. (4). From these we determined the RMS deviation Δ_{rms} from REF,

$$\Delta_{\text{rms}} := \left[\left\langle (\sigma_s - \sigma_s^{\text{ref}})^2 \right\rangle_s \right]^{1/2}.$$

The results for the seven algorithms using the 1-fs step size are shown in Figure 9. Note that the observed deviation are not exclusively caused by algorithmic approximations: due to the limited trajectory length, statistical fluctuations contribute to the observed Δ_{rms}. To estimate that statistical contribution, two reference simulations were compared with each other. The leftmost column in Figure 9 shows this estimate; its value is marked by the dashed line. Only deviations, which clearly exceed that value, as is the case for DC-i and DC-0, point toward algorithmic artifacts. For the other algorithms the observed deviations are insignificant.

One might expect that the large random forces caused by DC-i and DC-0 (cf. Fig. 3) should increase atomic mobilities. However, a closer analysis of the data (not shown) reveals that these two methods actually *decrease* the σ_s. We attribute this effect to a decorrelation of atomic motions due to the algorithmis random forces, resulting in a reduction of the inertial character and, hence, of the amplitude of these motions.

FIGURE 8. Low-frequency spectra of atomic mobilities; (top) spectrum obtained using REF (dashed), CUT (bold, solid), and DC-1d (thin, solid); (bottom) relative errors S^a/S^{ref} of the spectra S^a for the algorithms $a \in$ {CUT, DC-1d}.
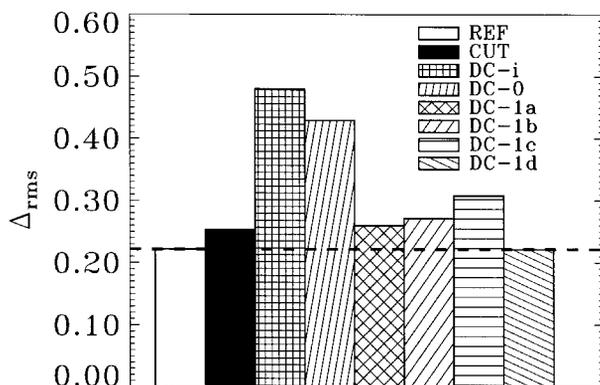
FIGURE 9. Root mean squared deviations of atomic mobilities from reference mobilities, computed for the seven different MD algorithms.

CORRELATION OF ATOMIC MOTIONS

This speculation can be readily verified by studying correlations of atomic motions. For that purpose, displacement autocorrelation functions C_s^{dis} were determined according to eq. (5). Averages over 20 atoms are shown in Figure 10. These autocorrelation functions, like Δ_{rms}, are subject to statistical fluctuations. To estimate the statistical error, autocorrelation functions were computed from two reference simulation (thin, solid lines).

According to the results in Fig. 10, the tested algorithms can be classified into three groups: in agreement with the speculation voiced above, the DC algorithms DC-i and DC-0 underestimate autocorrelations. The other algorithms DC-1a, DC-1b, DC-1c, and CUT entail slight overestimates. No significant deviation occurs for DC-1d. Thus, the latter algorithm showed the best performance in the tests of relevant observables considered so far.

Velocity autocorrelation functions were derived according to eq. (6). For the investigated algorithms there were no significant deviations from REF observed. This result is explained by the close relation of autocorrelation functions to spectra of atomic motions, which are Fourier transformed C^{vel}. The velocity autocorrelation functions we studied are determined by the physically relevant part of the high-frequency spectrum discussed above, which also did not exhibit any artifacts.

So far, we merely considered dynamical properties of single atoms but no interatomic relations. To check whether *correlated* atomic motion are described reliably by the various algorithms, cross-correlation functions $K_{s_1 s_2}$ for 190 atom pairs were computed according to eq. (7). In the eight pictures in Fig. 11, each of the diamonds represents one atom pair. The vertical axis measures the

FIGURE 10. Average displacement autocorrelation functions $\langle C^{\text{dis}}(t) \rangle_s$ of atomic positions; the results of two reference simulations (thin, solid lines) are plotted for an estimate of statistical fluctuations.
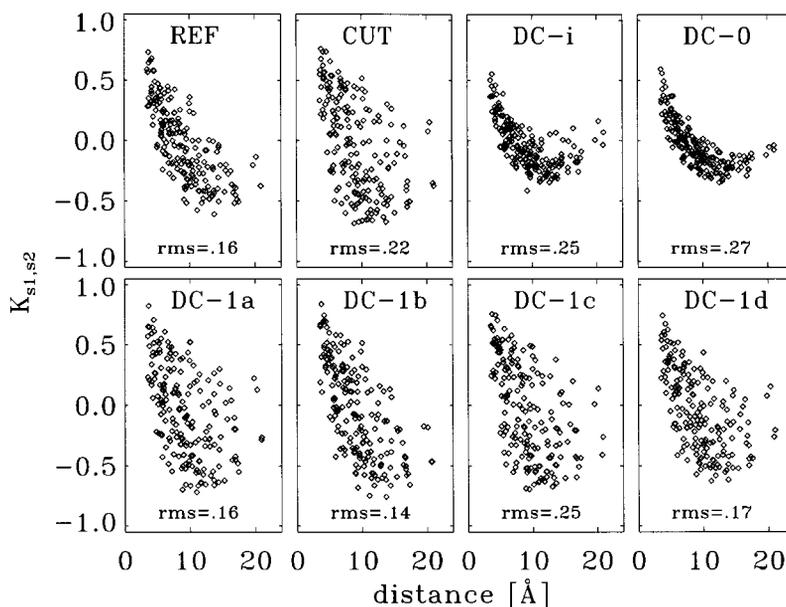
FIGURE 11. Cross-correlation functions of atomic positions; each of the 190 diamonds represents one atom pair; the vertical axis measures the value of its cross-correlation, the horizontal axis its average distance; in each of the eight pictures the overall RMSD of the cross-correlation from REF is indicated at the bottom; the RMSD between two reference simulations, indicated in the upper left picture of the figure, provides an estimate for the statistical error.

value of its cross-correlation, the horizontal axis its average distance. In addition, the overall RMS deviation (RMSD) of the cross-correlation from REF is indicated in each of the eight pictures. The RMSD between two reference simulations, indicated in the upper left picture of Figure 11, provides an estimate for the statistical error.

Inspection of the REF cross-correlations shows that in our protein model motions of close atoms tend to be correlated, whereas distant atoms exhibit anticorrelations. Whereas this behavior is qualitatively reproduced by all algorithms, quantitative differences can be readily observed: particularly at large distances, the noisy algorithms DC-i and DC-0 tend to suppress (anti-)correlations. Considering the overall RMSDs from REF, one observes that, in addition to DC-i and DC-0, CUT and DC-1c also provide cross-correlations that differ significantly from the reference values (although the scatterplots look almost identical). No significant deviations are seen for DC-1a, DC-1b, and DC-1d. Hence, cross-correlations provide an accuracy measure that is unable to discriminate these DC algorithms from REF. As we will show below, the projected configuration space densities introduced in the Methods Section provide a more sensitive measure.

PROJECTED CONFIGURATION SPACE DENSITY

In our particular choice for the conformational coordinates defined the projected configuration space densities ρ_c, we took care to monitor low-frequency motions of our protein model. To this aim we selected atoms, which in the tertiary structure of our model (cf. Fig. 2) exhibit large relative distances, for the definition of the conformational coordinates. Thus, the temporal evolution of these conformational coordinates should be dominated by collective motions of the backbone and, hence, should be of low-frequency character. As already noted in connection with Figure 8, such degrees of freedom are particularly influenced by long-range forces.

We determined $P = 50$ density distributions ρ_c for each of the two reference simulations and each of the seven tested MD algorithms. All comparisons of density distributions were carried out using nine different subdivisions $\{V^k\}$ of conformational spaces ranging from coarse grained ($K = 2$) to fine grained resolution ($K = 512$).

Additionally, a comparison of the two reference simulations served to estimate statistical errors

according to eqs. (8) and (9). The upper left picture of Figure 12 displays that estimate $\langle \Delta_{R_2 - R_1} \rangle$ (solid line). The horizontal axis measures the level K of graining. The increase of $\langle \Delta_{R_2 - R_1} \rangle$ with K merely reflects enhanced statistical fluctuations in the frequency counts of trajectory points within the conformation space volumes $\{V^k\}$. The standard deviation $\sigma_{R_2 - R_1}$ of our error estimate, determined by eq. (10), is plotted as dashed lines. The small deviation range indicates that, in our case, projected configuration space densities actually represent regular quantities.

The other seven pictures of Figure 12 show the averaged deviations Δ_a of projected phase space densities obtained by the seven MD algorithms a with respect to REF. Here, according to eq. (11), the statistical contribution $\langle \Delta_{R_2 - R_1} \rangle$ was subtracted from the observed deviations $\langle \Delta_{S_a - R_1} \rangle$. Two error ranges are depicted, a lower 2σ limit (dashed) obtained by eq. (12), as well as an upper limit (dashed-dotted) obtained by eq. (13).

In all cases, the most significant deviations are observed at intermediate levels K of coarse graining. These levels appear to provide a sufficiently detailed description of the conformation space densities ρ_c to allow an identification of algorithmic artifacts. In contrast, the coarse grained view on ρ_c at small K levels, as well as the increased statistical errors at large K, prevent a clearcut distinction of algorithms.

Highly significant deviations Δ_a are observed for CUT, as well as for DC-1b. The most significant
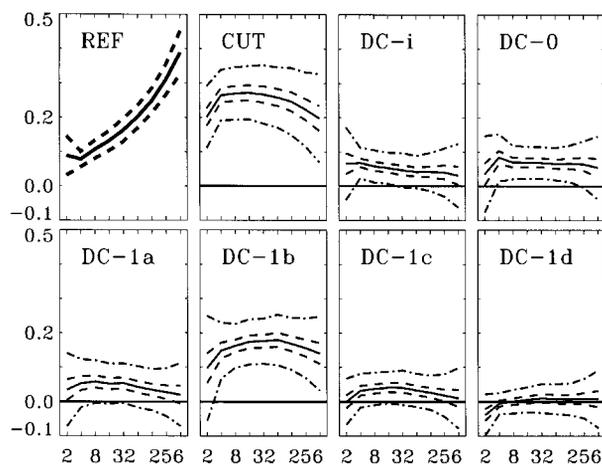
FIGURE 12. Algorithmic deviations of projected phase space densities and their statistical significance; the upper left graph provides an estimate of the statistical error; the horizontal axis measures the degree K of coarse graining of conformational space.

deviations for DC-i, DC-0, DC-1a, and DC-1c are smaller by a factor of 4−5 and approximately of size σ_a^{max}, which is 7 times the size of σ_a^{min}. To comment on the statistical significance of the latter result, we note that the error range σ_a^{max} represents an estimate for the worst case only, in which all P density distributions employed for the evaluation of Δ_a are *completely* correlated. Therefore, one may assume that the actual σ_a is smaller than σ_a^{max} and we may attribute statistical significance to the ρ_c deviations of DC-i, DC-0, DC-1a, and DC-1c, too. If we consider $\sigma_a^{max}/2$ a reasonably safe estimate for σ_a, then the hypothesis that algorithmic artifacts are absent in those four cases can be rejected at a level of significance below 5%. DC-1d is the only method for which the deviation stays well even within the σ_a^{min} range; therefore, no significant artifact is detected.

Discussion

Aiming at an evaluation of MD methods particularly designed for a description of protein dynamics, we presented *problem-adapted* measures for algorithmic accuracy. A closer inspection of the principles of MD simulation led us to the proposal that such measures of accuracy should be based on comparisons of *regular*, on nonchaotic, and *relevant quantities*. These comparisons should be of type Δ_n: the quantities derived from test simulations employing the MD algorithm to be evaluated should be compared with "exact" reference quantities.

The requirement to compare *regular* quantities, which generally are ensemble or time averages and therefore show slow convergence demanding extended test simulations, enforced the use of a simplified protein model as a test system. This allowed us to separate purely statistical fluctuations from the deviations caused by algorithmic artifacts.

The requirement to use *relevant* quantities led us to select a set of quantities that are considered useful for descriptions of protein function on the basis of MD simulations. The chosen quantities included atomic fluctuations, vibrational spectra, and various correlation functions of atomic motions, as well as projections of configuration space density distributions. In addition, conservation of total energy representing a conventional, although in our view irrelevant, measure of accuracy was studied.

We applied these measures of accuracy to seven MD algorithms that all achieve a reduction of computational effort by different types of approximation for the long-range Coulomb interaction. The studied algorithms include a conventionally used cutoff method, as well as a variety of recently proposed DC algorithms, that belong to the family of MTS methods. For a majority of the integration steps, the latter replace exact long-range forces by extrapolations using forces computed at previous steps. They differ from each other by the respective extrapolation method.

As summarized in Table II, we observed that DC methods, particularly an algorithm denoted as DC-1d, reproduce the chosen relevant quantities more accurately than the cutoff method. Specifically, the latter algorithm was found to artificially suppress low-frequency atomic motions at frequencies around 0.1 ps^{-1} by up to 40%, in addi-

TABLE II.
Observed Artifacts for Relevant Observables.

Observable	CUT	DC-i	DC-0	DC-1a	DC-1b	DC-1c	DC-1d
ΔE	—	••	••	•	•	•	•
S^{hf}	—	••	•	••	•	••	••
S^{lf}	••	—	—	—	—	—	—
σ_s	—	•	•	—	—	—	—
C_s^{dis}	•	••	••	•	•	•	—
C_s^{vel}	—	—	—	—	—	—	—
$K_{s_1 s_2}$	•	•	•	—	—	•	—
ρ_c	••	•	•	•	••	•	—

Total energy drift (ΔE), high frequency spectra (S^{hf}), low frequency spectra (S^{lf}), atomic fluctuations (σ_s), displacement autocorrelation (C_s^{dis}), velocity autocorrelation (C_s^{vel}), cross-correlation of atomic motions ($K_{s_1 s_2}$), and projected configuration space densities (ρ_c) are characterized by (—) no significant deviation from the reference simulation, (•) deviation observed, and (••) severe deviation; the three vertical lines group methods that show similar deviation patterns.

tion to the known structural artifacts caused by this method.[78] None of the DC methods showed that artifact.

By construction, DC methods exhibit a different type of artifact: as they approximate the long-range forces, they introduce small errors into the simulations. That algorithmic noise causes a significant drift of total energy, which was found to be larger by 3–5 orders of magnitude than the one present in an ''exact'' force computation. However, such energy drift can easily be suppressed by a weak coupling to a heat bath.[41, 79] Our test simulations served to address the question of whether the algorithmic noise forces and the necessary temperature correction cause relevant artifacts.

Two of the investigated DC methods, DC-i and DC-0, exhibited considerably larger noise than the others. We found that in these two methods noise decorrelates atomic motions, an effect that becomes apparent by inspecting autocorrelation functions, as well as cross-correlations.

As a particularly sensitive tool to measure algorithmic accuracy with respect to low-frequency collective motions, we studied projected configuration space densities. Because such quantities are directly related to statistical observables like entropy, free energy, or reaction rates, we considered them to be most relevant for protein function. With respect to the projected configuration space densities, DC-1d was the only approximation method that did not show any deviation from an exact force computation. In contrast, particularly large artifacts were detected for the cutoff method, as well as for algorithm DC-1b. The large deviations displayed by the latter method are somewhat surprising, because they are in contrast to the other measures of accuracy.

Considering the whole set of relevant quantities that we studied, DC-1d turned out to be the only MD algorithm that did not exhibit any algorithmic artifacts. Because DC methods are generally as efficient as the cutoff method, algorithm DC-1d should be preferred for protein dynamics simulations.

Table II also shows that the conventional criterion of total energy conservation yields a significantly different ranking of the seven algorithms considered. Accordingly, our problem-adapted approach employing regular and functionally relevant quantities also demonstrates that focusing on energy conservation may provide misleading information on the suitability of MD methos for protein dynamics simulations and therefore is inappropriate for that purpose. Generally, not a single observable, but rather the combined consideration of a number of relevant quantities, provides a comprehensive picture of algorithmic suitability.

Acknowledgments

References

1. M. Levitt, *J. Mol. Biol.*, **168**, 621 (1983).
2. H. Fraunfelder, *Biophys. J.*, **47**, 35 (1985).
3. J. A. McCammon and M. Karplus, *Proc. Natl. Acad. Sci. USA*, **76**, 3585 (1979).
4. M. Levitt and S. Lifson, *J. Mol. Biol.*, **46**, 269 (1969).
5. J. A. McCammon, B. R. Gelin, and M. Karplus, *Nature (Lond.)*, **267**, 585 (1977).
6. W. F. van Gunsteren and H. J. C. Berendsen, *Mol. Phys.*, **34**, 1311 (1977).
7. J. A. McCammon and S. C. Harvey, *Dynamics of Proteins and Nucleic Acids*, Cambridge University Press, Cambridge, U.K., 1987.
8. G. Petsko, *Nature (Lond.)*, **371**, 740 (1994).
9. H. J. C. Berendsen, *Science*, **271**, 954 (1996).
10. H. Heller, M. Schaefer, and K. Schulten, *J. Phys. Chem.*, **97**, 8343 (1993).
11. F. Zhou, A. Windemuth, and K. Schulten, *Biochemistry*, **32**, 2291 (1993).
12. O. Edholm, O. Berger, and F. Jähnig, *J. Mol. Biol.*, **250**, 94 (1995).
13. H. Grubmüller, B. Heymann, and P. Tavan, *Science*, **271**, 997 (1996).
14. M. Rief, F. Oesterhelt, B. Heymann, and H. E. Gaub, *Science*, **275**, 1295 (1997).
15. W. F. van Gunsteren and H. J. C. Berendsen, *Angew. Chem. Int. Ed. Engl.*, **29**, 992 (1990).
16. N. S. Ostlund and R. A. Whiteside, *Ann. N.Y. Acad. Sci.*, **439**, 195 (1985).
17. H. Heller, H. Grubmüller, and K. Schulten, *Mol. Simul.*, **5**, 133 (1990).
18. B. R. Brooks and M. Hodošček, *Chem. Design Autom. News*, **7**, 16 (1992).
19. B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus, *J. Comput. Chem.*, **4**, 187 (1983).
20. W. F. van Gunsteren and H. J. C. Berendsen, *GROMOS Manual*, BIOMOS b.v., Biomolecular Software, Groningen, The Netherlands, 1987.
21. B. Leimkuhler and R. D. Skeel, *J. Comput. Phys.*, **112**, 117 (1994).
22. B. Leimkuhler, S. Reich, and R. D. Skeel, In IMA Volumes in Mathematics and Its Applications, Vol. 82, K. Schulten and J. P. Mésivav, Eds., Springer, New York, 1996, p. 161.

23. R. D. Skeel, G. H. Zhang, and T. Schlick, *SIAM J. Sci. Comput.*, **18**, 203 (1997).

24. A. Ahmad and L. Cohen, *J. Comput. Phys.*, **12**, 389 (1973).

25. W. B. Streett, D. J. Tildesley, and G. Saville, *Mol. Phys.*, **35**, 639 (1978).

26. R. C. Y. Chin, G. W. Hedstrom, and F. A. Howes, *Considerations on Solving Problems with Multiple Scales*, 1st ed., Academic Press, Orlando, FL, 1985, p. 1.

27. A. Windemuth, Diploma thesis, Technical University of Munich, Munich, 1988.

28. M. E. Tuckerman, G. J. Martyna, and B. J. Berne, *J. Chem. Phys.*, **93**, 1287 (1990).

29. H. Grubmüller, H. Heller, A. Windemuth, and K. Schulten, *Mol. Simul.*, **6**, 121 (1991).

30. M. E. Tuckerman and B. J. Berne, *J. Chem. Phys.*, **94**, 1465 (1991).

31. M. E. Tuckerman, B. J. Berne, and G. J. Martyna, *J. Chem. Phys.*, **94**, 6811 (1991).

32. R. D. Skeel, J. J. Biesiadecki, and D. Okunbor, In *Proceedings of the International Conference on Computation of Differential Equations and Dynamical Systems*, World Scientific, Singapore, 1992.

33. R. D. Skeel and J. J. Biesiadecki, *Ann. Num. Math.*, **1**, 191 (1994).

34. H. Grubmüller, Ph.D. thesis, Technische Universität, München, 1994.

35. D. Okunbor and R. D. Skeel, *Explicit Canonical Methods for Hamiltonian Systems* [Working document], Numerical Computing Group, University of Illinois at Urbana–Champaign, 1991.

36. A. Windemuth, *Advanced Algorithms for Molecular Dynamics Simulation: The Program PMD*, ACS Books, Washington, D.C., 1995.

37. D. D. Humphreys, R. A. Friesner, and B. J. Berne, *J. Phys. Chem.*, **99**, 10674 (1995).

38. P. Procacci, T. Darden, and M. Marchi, *J. Phys. Chem.*, **100**, 10464 (1996).

39. S. J. Stuart, R. Zhou, and B. J. Berne, *J. Chem. Phys.*, **105**, 1426 (1996).

40. R. Zhou and B. J. Berne, *J. Phys. Chem.*, **103**, 9444 (1995).

41. T. Schlick, E. Bartha, and M. Mandziuk, *Annu. Rev. Biophys. Biomol. Struct.*, **26**, 181 (1997).

42. A. W. Appel, *SIAM J. Sci. Stat. Comput.*, **6**, 85 (1985).

43. J. Barnes and P. Hut, *Nature (Lond.)*, **324**, 446 (1986).

44. L. Greengard and V. Rokhlin, *Chem. Scr.*, **29A**, 139 (1989).

45. J. F. Leathrum and J. A. Board, *The Parallel Fast Multipole Algorithm in Three Dimensions* [Technical report], Dept. of Electrical Engineering, Duke University, Durham, NC, 1992.

46. C. Niedermeier and P. Tavan, *J. Chem. Phys.*, **101**, 734 (1994).

47. C. Niedermeier, Ph.D. thesis, Ludwig-Maximilians-Universität, München, Germany, 1995.

48. C. Niedermeier and P. Tavan, *Mol. Simul.*, **17**, 57 (1996).

49. B. A. Luty, I. G. Tironi, and W. F. van Gunsteren, *J. Chem. Phys.*, **103**, 3014 (1995).

50. U. Essmann, L. Perera, M. L. Berkowitz, T. Darden, H. Lee, and L. G. Pedersen, *J. Chem. Phys.*, **103**, 8577 (1995).

51. B. A. Luty, I. G. Tironi, and W. F. van Gunsteren, *J. Chem. Phys.*, **103**, 3014 (1995).

52. B. A. Luty and W. F. van Gunsteren, *J. Phys. Chem.*, **100**, 2581 (1996).

53. S. J. Aarseth, *Direct Methods for N-Body Simulations*, 1st ed., Academic Press, Orlando, FL, 1985, p. 378.

54. H. Grubmüller, Diploma thesis, Technische Universität München, Germany, 1989.

55. M. E. Tuckerman and B. J. Berne, *J. Chem. Phys.*, **95**, 8362 (1991).

56. L. Greengard and V. Rokhlin, *J. Comput. Phys.*, **73**, 325 (1987).

57. M. Eichinger, H. Grubmüller, H. Heller, and P. Tavan, *J. Comput. Chem.*, **18**, 1729 (1997).

58. L. Verlet, *Phys. Rev.*, **159**, 98 (1967).

59. G. D. Quinlan and S. Tremaine, *Astron. J.*, **100**, 1694 (1990).

60. K. D. Gibson and H. A. Scheraga, *J. Comput. Chem.*, **11**, 468 (1990).

61. M. Prévost, D. van Belle, G. Lippens, and S. Woodak, *Mol. Phys.*, **71**, 587 (1990).

62. R. J. Loncharich and B. R. Brooks, *Proteins*, **6**, 32 (1989).

63. H. Grubmüller and P. Tavan, *J. Chem. Phys.*, **101**, 5047 (1994).

64. A. Windemuth and K. Schulten, *Mol. Simul.*, **5**, 353 (1991).

65. R. D. Skeel, private communication, 1991.

66. R. D. Skeel and J. J. Biesiadecki, In *Proceedings of SCADE93*, Univ. of Auckland, New Zealand, January 4–8, 1993.

67. H. Frauenfelder, S. G. Sligar, and P. G. Wolynes, *Science*, **254**, 1598 (1991).

68. W. Doster, S. Cusack, and W. Petry, *Nature (Lond.)*, **337**, 754 (1989).

69. H. Frauenfelder, N. A. Alberding, A. Ansari, D. Braunstein, B. R. Cowen, M. K. Hong, I. E. T. Iben, J. B. Johnson, S. Luck, M. C. Marden, J. R. Mourant, P. Ormos, L. Reinisch, R. Scholl, A. Schulte, E. Shymasunder, L. B. Sorensen, P. J. Steinbach, A. Xie, R. D. Young, and K. T. Yue, *J. Phys. Chem.*, **94**, 1024 (1990).

70. J. Zollfrank, J. Friedrich, and F. Parak, *Biophys. J.*, **61**, 716 (1992).

71. W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C*, Cambridge University Press, Cambridge, U.K., 1988.

72. W. Nadler, A. T. Brünger, K. Schulten, and M. Karplus, *Proc. Natl. Acad. Sci. USA*, **84**, 7933 (1987).

73. H. Frauenfelder, P. J. Steinbach, and R. D. Young, *Chem. Scr.*, **29A**, 145 (1989).

74. H. Grubmüller, *Phys. Rev. E*, **52**, 2893 (1995).

75. P. A. Hargrave, H. E. Hamm, and K. P. Hofmann, *BioEssays*, **15**, 43 (1993).

76. K. Schulten and M. Tesch, *Chem. Phys.*, **158**, 421 (1991).

77. R. A. Marcus and N. Sutin, *Biocheim. Biophys. Acta*, **811**, 265 (1985).

78. C. L. Brooks III, B. M. Pettitt, and M. Karplus, *J. Chem. Phys.*, **83**, 5897 (1985).

79. H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, A. DiNola, and J. R. Haak, *J. Chem. Phys.*, **81**, 3684 (1984).

80. O. Gonzalez and J. Simo, *Comput. Methods Appl. Mech. Eng.*, **134**, 197 (1996).